REMARKS

Claims 1-4 and 6-19 are pending. Claims 1, 6, 10 and 13 are independent. Claims 1, 6-8 and 10-12 are amended and claims 13-19 are new.

Allowable Subject Matter

Applicant wishes to thank the Examiner for the indication that claims 4 and 8 contain allowable subject matter and would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claim Rejection - § 112

Claims 1-12 have been rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Applicant respectfully traverses this rejection.

As per the Examiner's comments, claims 1, 6-8 and 10 have been amended. Therefore, Applicant respectfully requests removal of this rejection.

Claim Rejection - § 102(b)

Claims 1, 2, 5-7 and 10-12 have been rejected under 35 U.S.C. § 102(b) as being clearly anticipated by Hugard et al. (U.S. Patent No. 5,745,667). Applicant respectfully traverses this rejection.

Hugard relates to a system and method for monitoring, restoring and recovering from configuration changes of a computer system. In particular, Hugard contains a system or tool to monitor configuration changes to recover working configuration from a stored configuration in a HDD disk when a change has occurred that causes an error. Further, a recovery utility is provided to support recovering the system when the configuration causes a faulty or improper operation. A configuration recovery tool detects whether a file related to system configuration information is changed, determines whether it is necessary to create a copy of the system configuration information and assists in determining whether to undo or correct changes of the system configuration.

As a method of monitoring and restoring system configuration, Hugard et al. disclose the following. First, a backup copy of computer system configuration is stored in a hard disk drive. After the first backup copy is created, the selected exiting system configuration files and the first backup copy are compared to determine whether or not there are any changes whenever the computer is powered on and started (the configuration files scanned when starting up are autoexec.bat, config.sys, system.ini). Second, if there are any changes, a menu screen is displayed. The menu screen enables a user to verify whether the computer system works properly under the current computer system configuration.

Next, if the computer system works properly, a menu screen is displayed to a user to maintain the second backup configuration with contents changed from the first backup copy. If the computer system works improperly when the computer is restarted with the changed configuration, a recovery utility capable of undoing the contents of the updated second backup copy is provided for displaying a menu screen to recover to the contents of the first backup copy, and supporting the computer system to work properly.

Hugard relates to enhancing system stability through operation of stored configuration files (autoexec.bat, configu.sys, system.ini.) by a recovery tool and recovering the stored initial files if changed files might operate the system improperly. Therefore, Hugard et al. do not relate to, "a method for fast booting a computer system," as recited by claims 1 and 10.

Further, Hugard et al. merely check stored data to determine if there are any errors. Therefore, Hugard et al. do not disclose, "checking whether a boot configuration information including a system booting state which was created while executing a previous normal booting process exists or not," as recited in part by claims 1 and 10 as amended.

Accordingly, claims 1 and 10 are allowable over the prior art. Regarding dependent claims 2, 6-7 and 11-12, these claims are allowable over the prior at for at least the same reasons as their corresponding independent claims. Therefore, Applicant respectfully requests removal of this rejection.

Claims 3 and 9 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Hugard et al. Applicant respectfully traverses this rejection.

As discussed above, Hugard et al. fail to disclose all of the features of independent claim 1 from which claims 3 and 9 depends. Therefore, claims 3 and 9 are allowable over the prior art for at least the same reasons as their corresponding independent claim 1, and Applicant respectfully requests removal of this rejection.

New Claims

Newly added claims 13-19 are supported in the specification on at least pages 8-13. Further, claim 13 is allowable over the prior art because the prior art does not disclose "a method for supporting fast booting a computer system through storing/resuming a memory status of the system, comprising the steps of: checking whether to store a memory contents status; checking memory contents of a certain unit of the memory; selectively storing contents written in an area necessary for system operation based on the memory contents checking result; and resuming the stored contents for fast booting," as recited by claim 13.

Regarding dependent claims 14-19, these claims are allowable for at least the same reasons as corresponding independent claim 13.

CONCLUSION

In view of the above amendments and remarks, reconsideration of the rejection and allowance of claims 1-4 and 6-19 is respectfully requested.

Attached hereto is a marked-up version of the changes made to the application by this Amendment.

If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is invited to contact the

Jayne Saydah (Reg. No. 48,796) at (703) 205-8000, in the Washington, D.C. area.

Applicant respectfully petitions under the provisions of 37 CFR 1.136(a) and 1.17 for a one-month extension of time in which to respond to the Examiner's Office Action. The Extension of Time Fee in the amount of $110.00 is attached hereto.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17; particularly, extension of time fees.

Respectfully submitted,

BIRCH, STEWART, KOLASCH & BIRCH, LLP

By James T. Eller, Jr.
James T. Eller, Jr.
Reg. No. 39,538

JTE/JES:sld
ns

P.O. Box 747
Falls Church, VA 22040-0747
(703) 205-8000

Attachment: Abstract
Version With Markings Showing Changes Made

VERSION WITH MARKINGS SHOWING CHANGES MADE

IN THE ABSTRACT

The Abstract has been rewritten as follows.

ABSTRACT

[This invention provides] A method for quickly booting a personal computer system **[by]** using **[a]** boot configuration information on memory and the attached devices that was created and saved in a hard disk at the preceding boot process. The method for a quick boot process **[according to the present invention comprises]** **includes** the steps of performing a power-on self test (POST) operation when a personal computer system is powered on or a reset button is pressed; performing a normal boot process after the POST operation; saving the contents of memory and the status of the attached devices to a hard disk; checking if a reboot is requested; restoring the saved boot configuration information from the hard disk, after POST is completed during the reboot process; checking whether or not an initial device configuration file and/or an automatic batch file were changed; and executing commands in the two files and saving a newly created boot configuration information to the hard disk for future boot. **[Accordingly, this invention enable to boot a]** The personal computer system **may reboot** quickly because of omission of execution of the initial device configuration filed and the automatic batch file.

IN THE SPECIFICATION

The paragraph beginning on page 1, line 23, and continuing to page 2, has been rewritten as follows:

--Figure 1 shows a block diagram of the system architecture for a conventional personal computer system, comprising a central processing unit (CPU) 1; a read only memory (ROM) 2 for permanent storage of basic input output system (BIOS) and the initial states of internal devices, a random access memory (RAM) 3 for temporary storage of information; a micro computer (MICOM) 4 for controlling peripheral devices such as a keyboard input device, a mouse input device, and a power supply **7**; a hard disk (HDD) 8 for providing a secondary information storage; a disk controller 5 for controlling HDD; a video output display 6 for displaying information; and a power supply 7. When power is applied to the computer system, the computer system starts to be booted to load an operating system (OS) and thus is brought into a known useful state in which application programs can be executed. This procedure is generally called “booting”. An operating system is a software that provides resource management on a computer system, including basic tasks such as process execution, memory management, and file management. Examples are MS-DOS, Windows95, OS/2, and UNIX. Execution of user applications is based on these basic functions of the operating system.--

The paragraph beginning on page 2, line 19, and continuing to page 3, has been rewritten as follows:

--The boot process of an IBM PC in which MS-DOS operating system is already installed is as follows. When a user turns the personal computer power switch on or presses a reset button, a power-on self test (POST) is performed by ROM BIOS codes to diagnose each component of the personal computer. Next, a file called MSDOS.SYS is loaded and executed, and another file called IO.SYS is then loaded and executed to perform certain preliminary functions related to management of such peripheral devices as keyboard, disk, and display. And then, a command preprocessor or COMMAND.COM is loaded into a memory that receives, interprets and executes user commands. A file called CONFIG.SYS that specifies devices possibly connected to the personal computer is loaded and ASCII statements contained therein are executed to load device drivers and initialize them. Finally, another ASCII file called AUTOEXEC.BAT is loaded and then programs that **[is]** **are** listed therein are executed, thereby preparing the personal computer for use.--

The paragraph beginning on page 3, line 11, has been rewritten as follows:

--There two kinds of boots; "cold boots" and "warm boots", which rely on the state of the computer system when the boot operation is requested. A "cold boot" is performed when power is applied to **the** computer or a reset button is pressed. When an operating system is loaded in memory already and the computer system is powered on already, a user may request a "warm boot" by entering a predefined sequence of key strokes, e.g., <Ctrl>+<Alt>+<Del>. The BIOS codes **[consist of]** **include** a plurality of computer routines for controlling devices such as **a** system clock, video output display **6**, disk **[driver]** **controller** **5**, and keyboard and thus provide a low-level interface to these devices. The BIOS is generally stored in a Flash ROM.

The paragraph beginning on page 5, line 25, and continuing to page 6, has been rewritten as follows:

--It is therefore a primary object of the present invention to provide a method and apparatus that significantly reduces the time required for boot process after a POST operation by using a boot configuration information on memory and the attached devices that **[was]** **were** created and saved in a disk in the preceding boot process, and thereby skipping execution of statements in an initial device configuration file and an automatic batch file.--

IN THE CLAIMS

The claims have been amended as follows:

1. A method for **fast** booting a computer system, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. checking whether a boot configuration information **including a system booting state** which was created **[in the preceding]** **while executing a previous normal** booting process exists or not;

C. storing **[a]** **the** boot configuration information **from** **[after]** execution of the POST operation **[to a disk]** **before loading a graphic interface (GUI) program**, based on the checking result; and

D. loading **[a]** **the** graphic user interface (GUI) program.

6. A method for **fast** booting a computer system, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. **[restoring]** **resuming** a boot configuration information **including a system booting state** by using the boot configuration information **[which has been stored after POST operation in a disk]** **which was stored while executing a previous normal boot process**; and

C. loading a graphic user interface (GUI) program.

7. (Amended) A method according to claim 6, wherein said step B further compris**es[ing]** the steps of:

checking if a designated boot configuration information is different from the **[restored]** **resuming** boot configuration information;

executing an initial driving program based on a modified configuration information; and updating the boot configuration information after said execution.

8. (Amended) A method according to claim 6, wherein said step B compris**[ing]es** the steps of:

determining whether to **[restore]** **resume** said stored boot configuration information;

[restoring the] **resuming the** contents of memory blocks, addresses of which have been stored **while executing a previous normal booting process** **[in said disk]**; and

writing zeros into other memory blocks than the **[restored]** **resumed** memory blocks.

10. (Amended) A method for quickly booting a computer system in which Windows operating system is installed, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. checking whether a boot configuration information **including a system booting state** which was created **[in the preceding]** **while executing a previous normal** boot**ing** process exists or not;

C. storing the current boot configuration information **[in a disk storage medium]**, if there is no stored boot configuration information;

D. performing **[a quick]** **the** POST operation when the computer system is rebooted;

E. **[restoring]** **resuming** the stored boot configuration information **[from the disk storage medium]**; **and**

F. updating the boot configuration information before a graphic user interface (GUI) program is loaded, if a designated boot configuration information is different from the **[restored]** boot configuration information**[;].**

11. (Amended) A method according to claim 10, wherein said step B calls an interrupt for bootstrap loader to check if the boot configuration information which was created **[in the preceding]** **while executing a previous normal booting process** **[boot process exists]**.

12. (Amended) A method according to claim 10, wherein said step F determines whether or not the designated boot configuration information is different from the **[restored]** **resumed** boot configuration information based on changes of CONFIG.SYS file and/or AUTOEXEC.BAT file.

Claims 13-19 have been added.

Claim 5 has been canceled.